AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently amended) A system for incrementally executing a client/server application~~, leveraging existing communications network infrastructure having~~ the client/server application executed on ~~at least one~~a client ~~computer~~ and ~~at least one~~a server ~~computer~~, ~~wherein~~ the ~~at least one~~ client ~~computer~~ and the ~~at least one~~ server ~~computer are~~ in communication ~~with each other~~ over ~~one or more~~a communication link~~communications links within the network infrastructure~~, the system comprising:

a client/server application comprising a server application component and a client component, the server application component comprising a plurality of server component portions, and provided on the ~~at least one~~ server ~~computer~~, the server component portions including an initial server component portion and one or more subsequent server component portions, the initial server component portion having an initial function, each of the one or more subsequent server component portions including ~~at least one~~ a respective subsequent function;

the client application component provided on the ~~at least one~~ client ~~computer~~, the client component including ~~one or more~~a command selector~~s~~, ~~each of~~ the ~~one or more~~ command ~~selectors~~ selector having:

~~associated code~~means for selecting a function available from the plurality of the server component portions; and

means~~associated code~~ for generating a parameter for use by the server application component in determining the appropriate server component portion of the plurality of server component portions to load and execute on the server ~~computer~~ to provide the selected function to the client ~~computer~~, the parameter being an initial parameter or a subsequent parameter;

the ~~at least one~~ server ~~computer~~ including:

a memory for executing the server application component;

means for receiving an initiating message from the client ~~computer~~, the initiating message including the initial parameter associated with the initial server component portion;

means for loading into the memory, ~~in dependence upon~~in response the initial parameter, the initial server component portion;

means for executing the initial server component portion loaded into the memory to provide the initial function to the client ~~computer~~;

means for receiving a subsequent message from the client ~~computer~~, the subsequent message including the subsequent parameter associated with a subsequent server component portion;

means for loading into the memory, in response ~~in dependence upon~~ to the subsequent parameter, the subsequent server component portion; and

means for executing the subsequent server component portion loaded into the memory to provide the subsequent function to the client ~~computer~~.

2. (Currently amended) The system according to claim 1, wherein one ~~portion~~ of the plurality of server component portions is a compact portion loaded and executed on the ~~at least one~~ server ~~computer~~ upon receipt of a first application function request from the client component, the compact portion delivering a subset of functions applicable to commands most commonly requested to provide a fast executing initial portion of the application.

3. (Currently amended) A method for incrementally executing a client/server application, the application executed on ~~leveraging existing communications network infrastructure having~~ a ~~at least one~~ client ~~computer~~ and a ~~at least one server computer~~, ~~the at least one~~ server ~~computer~~ comprising a server processor and a memory for executing a server component wherein the ~~at least one~~ client ~~computer~~ and the ~~at least one~~ server ~~computer~~ are in communication ~~with each other~~ over ~~one or more~~ a communication link ~~communications links within the network infrastructure~~, the method comprising the steps of:

(i) providing a server application component comprising a plurality of server component portions on the ~~at least one~~ server ~~computer~~, the server component portions including an initial server component portion and one or more subsequent server component portions;

(ii) providing a client component on the ~~at least one~~ client ~~computer~~, the server application component and the client component forming ~~a~~ the client/server application;

(iii) loading the initial server component portion into the memory of the server ~~computer~~, in ~~dependence upon~~response to a parameter ~~contained~~ included in a request from the client component for an application function, ~~the initial server component portion~~;

(iv) executing, ~~on the server computer,~~ the initial server component portion to provide an initial function to the client ~~computer~~;

(v) loading into the memory of the server ~~computer~~, in ~~dependence upon~~response to a subsequent parameter ~~contained~~ included in a subsequent request from the client component, the subsequent server component portion; and

(vi) executing, ~~on the server computer,~~ the subsequent server component portion to provide a subsequent function to the client ~~computer~~.

4. (Currently amended) The method according to claim 3, wherein one ~~portion~~ of the plurality of server component portions is a compact portion loaded and executed on the server ~~computer~~ upon receipt of a first application function request from the client component, the compact portion delivering a subset of functions applicable to commands most commonly requested to provide a fast executing initial portion of the application.

5. (Currently amended) A system for incrementally executing a client/server application, the application executed on ~~leveraging existing communications network infrastructure having at least one~~a client ~~computer~~ and ~~at least one~~a server ~~computer~~, wherein the ~~at least one~~ client ~~computer~~ and the ~~at least one~~ server ~~computer~~ are in communication ~~with each other~~ over a communication link~~one or more communications links within the network infrastructure~~, the system comprising:

a module for providing a server application component comprising a plurality of server component portions on the ~~at least one~~ server ~~computer~~;

a module for providing a client component on the ~~at least one~~ client ~~computer~~, the server application component and the client component forming ~~a~~ the client/server application;

a module for loading and executing~~, on the server computer,~~ an appropriate server component portion from the plurality of server component portions ~~applicable to and upon~~in response to an initial request from the client component for an application function, the initial request including a parameter associated with the appropriate server component portion, the parameter being created at the client ~~computer~~; and

a module for loading and executing~~, on the server computer~~ an additional appropriate server component portion of the plurality of server component portions for an additional request received from the client component for an application function not available from any running server component portion, the additional request including a parameter associated with the additional appropriate server component portion, the parameter in the additional request being created at the client ~~computer~~.

6. (Currently amended) The system according to claim 5, wherein one ~~portion~~ of the plurality of server component portions is a compact portion loaded and executed on the server ~~computer~~ upon receipt of a first application function request from the client component, the compact portion delivering a subset of functions applicable to commands most commonly requested to provide a fast executing initial portion of the application.

7. (Currently amended) A storage medium readable by a computer encoding a computer program for execution by the computer to carry out~~, the medium encoding a computer process to provide~~ a method for incrementally executing a client/server application, the application executed on ~~leveraging existing communications network infrastructure having at least one~~a client ~~computer~~ and ~~at least one~~a server ~~computer~~, wherein the ~~at least one~~ client ~~computer~~ and the ~~at least one~~ server ~~computer~~ are in communication ~~with each other~~ over a communication link~~one or more communications links within the network infrastructure~~, the computer ~~process~~ program comprising:

~~a processing portion~~code means for ~~providing~~ a server application component comprising a plurality of server component portions ~~on the at least one server computer~~;

code means ~~a processing portion~~ for ~~providing~~ a client component on the ~~at least one~~ client ~~computer~~, the server application component and the client component forming ~~a~~ the client/server application;

code means ~~a processing portion~~ for loading and executing~~, on the server computer~~, an appropriate server component portion from the plurality of server component portions ~~applicable to and upon~~ in response to an initial request from the client component for an application function, the initial request including a parameter associated with the appropriate server component portion, the parameter being created at the client ~~computer~~; and

code means ~~a processing portion~~ for loading and executing~~, on the server computer~~, an additional appropriate server component portion of the plurality of server component portions for an additional request received from the client component for an application function not available from any running server component portion, the additional request including a parameter associated with the additional appropriate server component portion, the parameter in the additional request being created at the client ~~computer~~.

8. (Currently amended) The storage medium according to claim 7, wherein one ~~portion~~ of the plurality of server component portions is a compact portion loaded and executed on the server ~~computer~~ upon receipt of a first application function request from the client component, the compact portion delivering a subset of functions applicable to commands most commonly requested to provide a fast executing initial portion of the application.

9. (Cancelled)

10. (Previously presented) The method according to claim 3 wherein the step of loading the subsequent server component portion is omitted if the subsequent function associated with the subsequent running portion is available from any running server component portion of the server component.

11. (Previously presented) The method according to claim 3 further comprising the step of:

creating the parameter at a client application.

12. (Currently amended) The method according to claim 3 further comprising the steps of:

at the client ~~computer~~, accepting from a user, an indication of an end session;

at the client ~~computer~~, generating~~, in dependence upon the indication of an end session,~~ an end session message;

at the server ~~computer~~, receiving the end session message; and

at the server ~~computer~~, terminating the execution and unloading ~~from the memory, in dependence upon the end session message, of~~ the server component portions.

13. (Previously presented) The system according to claim 1 wherein the loading of the subsequent server component portion is omitted if the subsequent function associated with the subsequent server component portion is available from any running server component portion of the server component.

14. (Currently amended) The system according to claim 1, further comprising a plurality of clients, wherein each of the plurality of clients ~~client computer~~ includes means for creating the initial parameter and the subsequent parameter at a client application.

15. (Currently amended) The system according to claim 1, wherein the client ~~computer~~ includes:

means for accepting from a user, an indication of an end session; and

means for generating~~, in dependence upon the indication of an end session,~~ an end session message,

and wherein the server ~~computer~~ includes:

means for receiving the end session message; and

means for terminating the execution and unloading ~~from the memory, in dependence upon the end session message, of~~ the server component portions.

16. (Currently amended) A method of executing an application in an environment comprising a server and ~~one or more than one~~a client, the server comprising a server processor and a memory for executing a server component, the server component including an initial server component portion and one or more subsequent server component portions, the client comprising a client component, the server component and the client component forming the application, the method comprising the steps of:

a) at the server ~~computer~~, receiving an initial message from the client, the initial message including an initial parameter associated with the initial server component portion, the initial parameter being created by a client application;

b) at the server, loading into the memory of the server, in ~~dependence upon~~response to the initial parameter, the initial server component portion;

d) at the server, executing the initial server component portion in the memory to provide an initial function to the client;

e) at the server, receiving a subsequent message from the client, the subsequent message including a subsequent parameter associated with the subsequent server component portion, the subsequent parameter being created by the client application;

f) at the server, loading into the memory of the server, in dependence upon the subsequent parameter, the subsequent server component portion; and

g) at the server, executing the subsequent server component portion in the memory to provide a subsequent function to the client.

17. (Previously presented) The method according to claim 16 wherein the step of loading the

subsequent server component portion is omitted if the subsequent function associated with the subsequent server component portion is available from any running server component portion of the server component.

18. (Previously presented) The method according to claim 16 further comprising the step of:
creating the initial parameter and the subsequent parameter at a client application.

19. (Currently amended) The method according to claim 16 further comprising the steps of.
at the client, accepting from a user, an indication of an end session;
at the client, generating~~, in dependence upon the indication of an end session,~~ an end session message;
at the server, receiving the end session message; and
at the server, terminating the execution and unloading ~~from the memory, in dependence upon the end session message, of~~ the server component portions.

20. (Currently amended) A ~~system for executing an application in an environment comprising: a~~ server for ~~one or more than one~~a client, the server ~~including~~ comprising:
a memory for executing a server component, the server component including an initial server component portion and one or more subsequent server component portions, the client comprising a client component, the server component and the client component forming ~~the~~ an application, ~~the server further including:~~
means for receiving an initial message from the client, the initial message including an initial parameter associated with the initial server component portion, the initial parameter being created by a client application;
means for loading into the memory of the server, in ~~dependence upon~~response to the initial parameter;
means for executing the initial server component in the memory to provide an initial function to the client;
means for receiving a subsequent message from the client, the subsequent message including a subsequent parameter associated with the subsequent server component portion, the subsequent parameter being created by the client application;
means for loading into the memory of the server, in response to ~~dependence upon~~ the subsequent parameter, the subsequent server component portion; and
means for executing the subsequent server component in the memory to provide a subsequent function to the client.

21. (Currently amended) The ~~system~~ server according to claim 20, wherein the loading of the subsequent server component portion is omitted if the subsequent function associated with the subsequent server component portion is available from any running server component portion of the server component.

22. (Currently amended) The ~~system~~ server according to claim 20, wherein the client includes means for creating the initial parameter and the subsequent parameter at a client application.

23. (Currently amended) The ~~system~~ server according to claim 20, wherein the client includes means for accepting from a user, an indication of an end session, and means for generating, in response to ~~dependence upon~~ the indication of an end session, an end session message, and wherein the server includes means for receiving the end session message, and means for terminating the execution and unloading from the memory, in response to ~~dependence upon~~ the end session message, ~~of~~ the server component portions.